

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

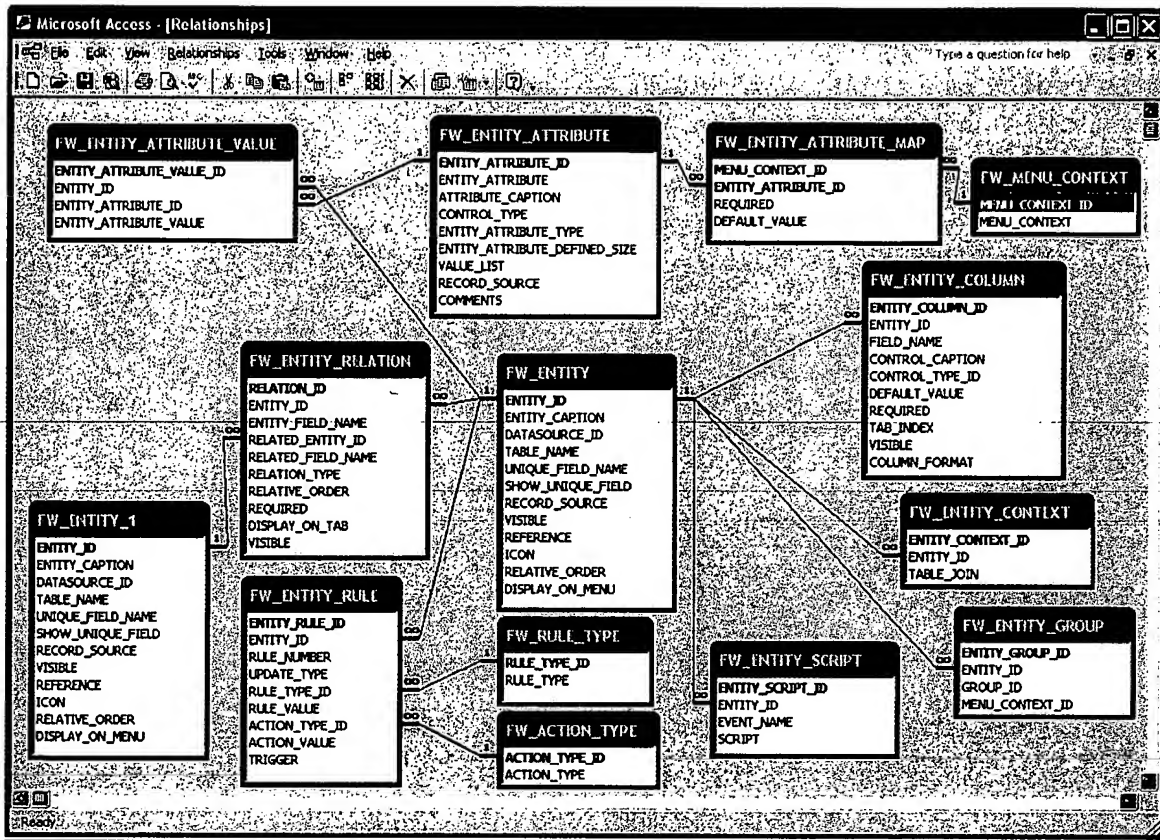
- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

As rescanning documents *will not* correct images,  
Please do not report the images to the  
Image Problem Mailbox.

# EXHIBIT D

## Entities ERD

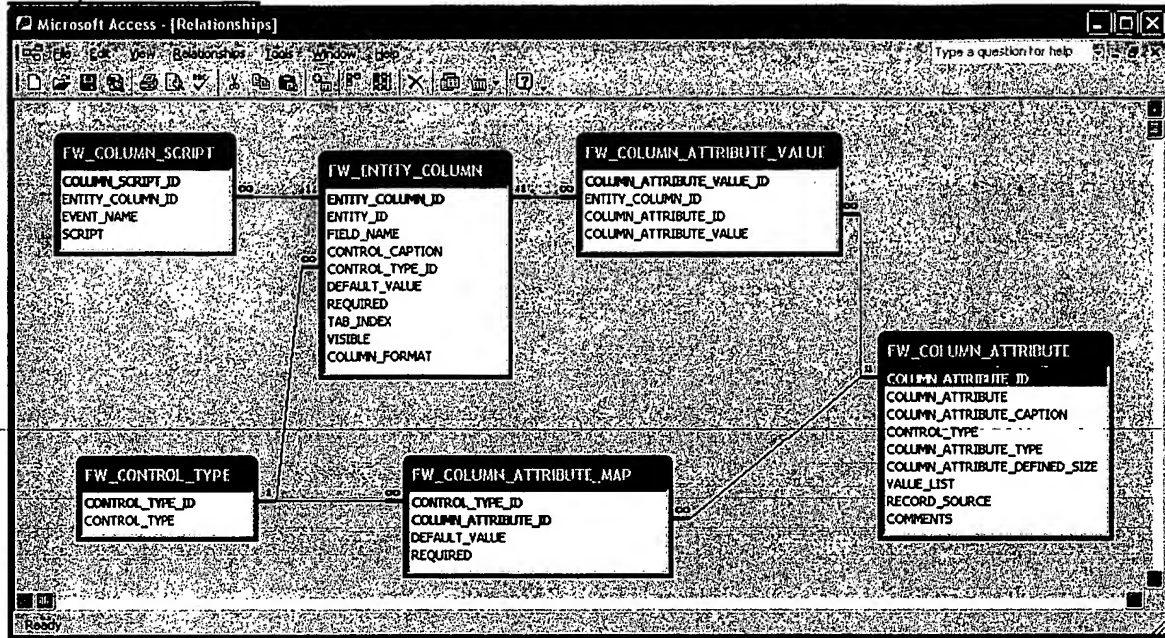


### Remarks

During the metadata generation process, the FW\_ENTITY table is derived from the underlying tables in the relational database. The FW\_ENTITY\_ATTRIBUTE table contains a predefined set of attributes that depend on the type of Entity, which is either a Parent Entity or Reference Entity. Depending on the Entity type, a different set of default attribute values are generated and stored in the FW\_ENTITY\_ATTRIBUTE\_VALUE table.

## EXHIBIT D

### Entity Fields ERD

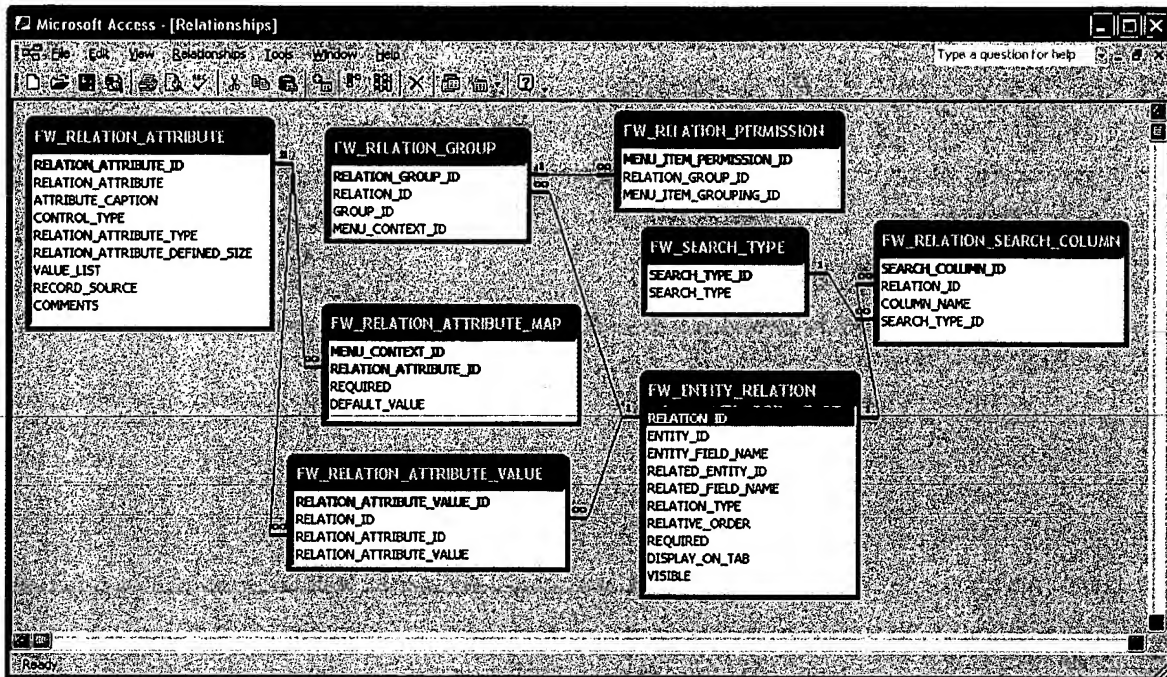


### Remarks

Entity Fields are derived from table columns in the underlying relational database schema. Default data entry controls are assigned to each Field depending on whether the column is a foreign key or, if not, depending on the datatype of the column. Once a data entry control has been assigned, an appropriate set of default attribute values are assigned to populate the FW\_COLUMN\_ATTRIBUTE\_VALUE table with metadata. For example, a combo box control would be assigned a set of attributes (defined in the FW\_COLUMN\_ATTRIBUTE table) that enable the user interface to generate a drop down list of values at run-time.

## EXHIBIT D

### Entity Relationships ERD

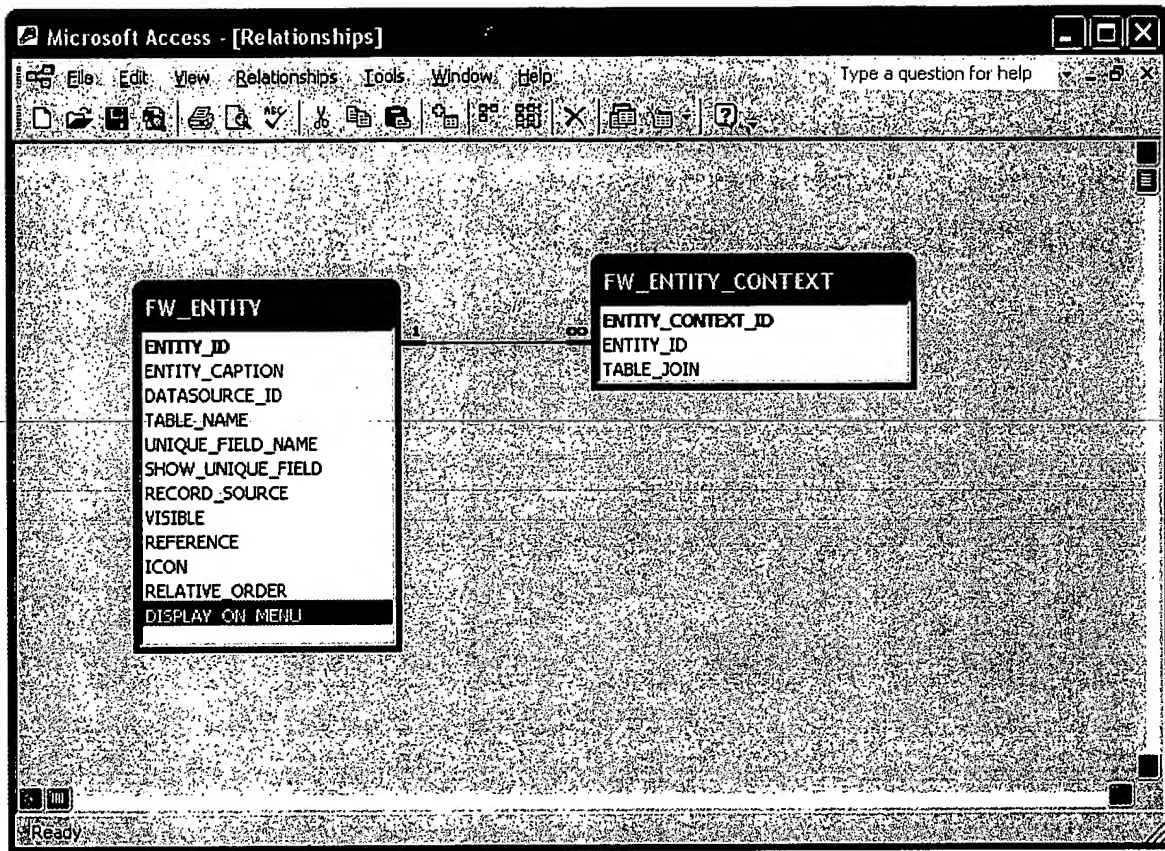


#### Remarks

Entity Relationships are derived from table foreign key information in the underlying relational database schema. The type of relationship between two tables determines the set of default attributes that are populated in the FW\_RELATION\_ATTRIBUTE\_VALUE metadata table. For each relationship found in the database schema, two entries are made in the FW\_ENTITY\_RELATION table, one for each side of the relationship (i.e. one-to-many and many-to-one).

## EXHIBIT D

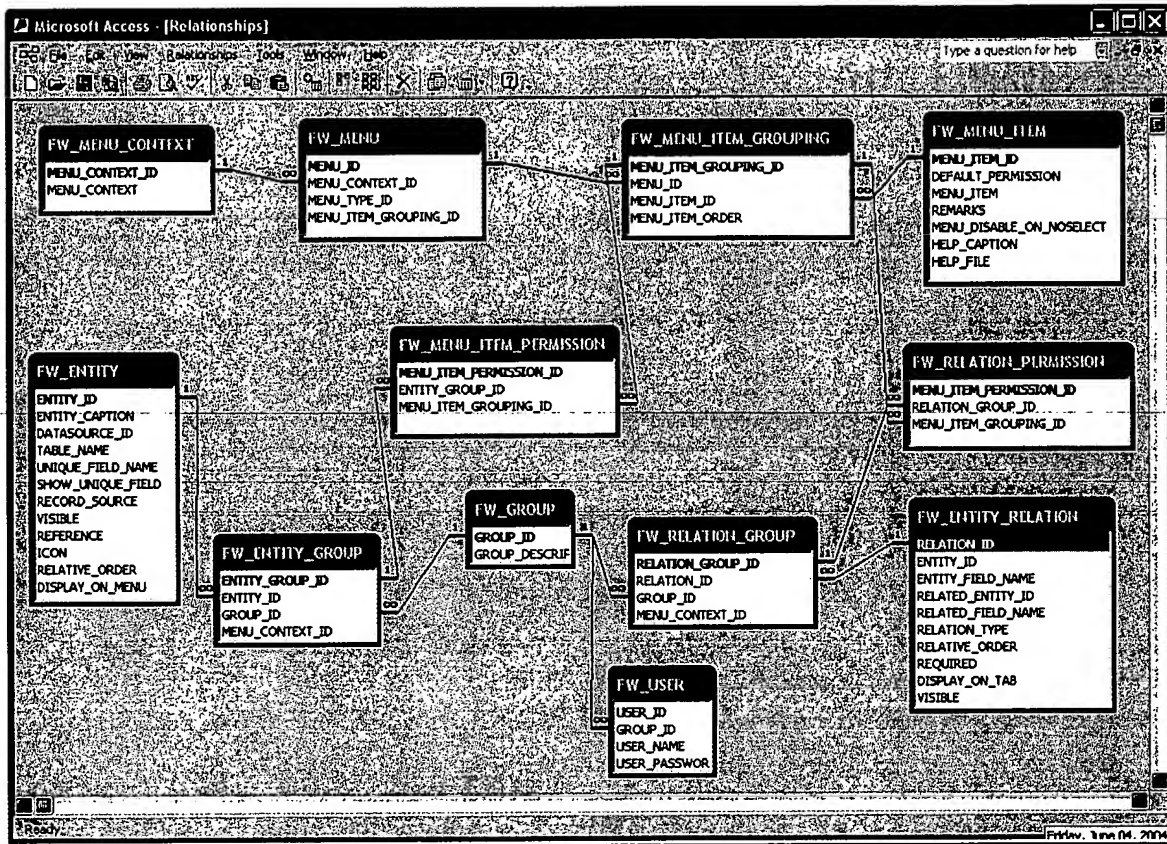
### Entity Search Path ERD



#### Remarks

To provide Advanced Searching functionality, predefined search paths are stored in the FW\_ENTITY\_CONTEXT metadata table. During the initial metadata generation process, FW\_ENTITY\_CONTEXT is populated with table joins that represent all tables directly related to the current entity. Additional joins can be added to a search path through the Utility Tool user interface. The complete set of table joins is then traversed at run-time to provide Advanced Searching functionality.

## Entity and Entity Relationship Permissions ERD

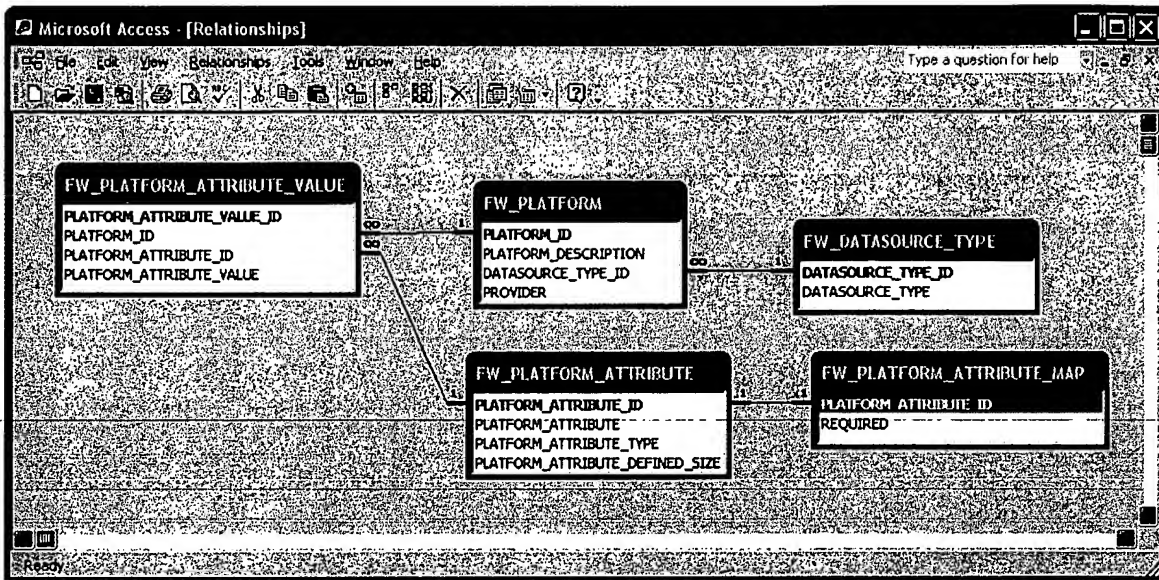
Remarks

As discussed in the patent application, permissions are assigned to context menu options. A different menu context is defined for each type of Entity (i.e. Parent or Reference) and each type of relationship (i.e. One-to-Many, Many-to-One, and Many-to-Many). The possible set of menu options available for each menu context is predefined by the metadata tables with the FW\_MENU prefix. During the initial metadata generation process, default permissions are set for the Administrators group. These permissions are set in the FW\_MENU\_ITEM\_PERMISSION (for Entities) and FW\_RELATION\_PERMISSION (for Entity Relationships). Additional permissions can be assigned to new groups of users through the Utility Tool user interface.



## EXHIBIT D

### Platform and Platform Attributes ERD



#### Remarks

The syntactic differences between various database platforms (Microsoft Access, Microsoft SQL Server, Oracle, etc) are predefined in the tables prefixed with FW\_PLATFORM. During the metadata generation process, a variety of SQL statements are automatically generated using the FW\_PLATFORM tables. This ensures that SQL appropriate to the underlying database platform is generated. At run-time, these same metadata tables facilitate generation of dynamic SQL appropriate to the underlying database platform.